



CRIQUET: un outil de base pour construire des systemes experts

Philippe Vignard

► To cite this version:

Philippe Vignard. CRIQUET: un outil de base pour construire des systemes experts. [Rapport de recherche] RR-0316, INRIA. 1984, pp.30. inria-00076241

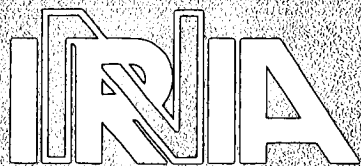
HAL Id: inria-00076241

<https://hal.inria.fr/inria-00076241>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



CENTRE
SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(3) 954 90 20

Rapports de Recherche

N° 316

CRIQUET
UN OUTIL DE BASE
POUR CONSTRUIRE
DES SYSTÈMES EXPERTS

Philippe VIGNARD

Juillet 1984

CRIQUET : un outil de base pour
construire des systèmes experts

Philippe VIGNARD +

+ INRIA, Sophia-Antipolis
06560 Valbonne FRANCE
Tel.: (93) 74-80-80
(poste 3305)

RESUME

L'objet de ce rapport est de décrire CRIQUET, un moteur d'inférence développé pour servir d'outil de base pour la construction de systèmes experts. Un module d'interface facilite la formulation des connaissances, spécifiées à l'aide de règles de production. Le système permet d'utiliser trois types de raisonnements et des tests de compatibilité. Les communications homme-machine sont facilitées.

ABSTRACT

This report describes CRIQUET, a production system for designing expert systems. A simple natural language understanding system facilitates the introduction and modification of production rules. Three types of search strategies including compatibility tests can be used. Of special interest are the man-machine communication facilities.



PLAN

----- Introduction

1/ Le moteur d'inférence	page 2
1-1/ Les constituants	page 2
1-2/ Types de connaissances exprimables	page 3
1-3/ Langage d'expression	page 3
1-4/ Faits et base de faits	page 4
1-5/ La base de règles	page 5
1-6/ Les mécanismes de raisonnement	page 7
1-6-1/ Le filtrage	page 7
1-6-2/ Les modes de raisonnement	page 7
a/ chaînage avant	page 7
b/ chaînage arrière	page 8
c/ chaînage mixte	page 9
1-6-3/ Choix d'une règle	page 10
1-7/ Méta-connaissances	page 11
1-8/ Le raisonnement approximatif	page 12
1-9/ La gestion de compatibilité	page 13
1-10/ Aspect interactif	page 15
2/ Le module d'interface	
2-1/ Introduction	page 16
2-2/ L'interface construite	page 16
2-3/ Principe de fonctionnement	page 16
2-4/ Enrichissement du vocabulaire	page 18
Conclusion	page 19
Annexes	page 20
Bibliographie	page 29

INTRODUCTION

La notion de systèmes experts recouvre de plus en plus de logiciels, dans de nombreux domaines fort divers. La méthodologie des systèmes experts a déjà été appliquée avec succès, par exemple, pour le diagnostic médical (MYCIN), l'interprétation de spectrogrammes de masse (DENDRAL), la classification géologique (PROSPECTOR) ou la reconnaissance de la parole (HEARSAYII) (12). Un moteur d'inférence nu, suffisamment puissant et général, peut faciliter ce développement dans tous les domaines où la modélisation des connaissances est possible à l'aide de règles de production.

Le moteur d'inférence CRIQUET est utilisé pour étudier une approche système expert pour la classification automatique des galaxies selon leur type morphologique (6). Une autre utilisation prévue de CRIQUET concerne l'élaboration d'une base documentaire mathématique intelligente.

I/ LE MOTEUR D'INFERENCE

I-1/ Les constituants de la base de connaissances

Comme dans un système de production (9), les 2 constituants classiques de la base de connaissances sont une base de règles et une base de faits. Les éléments composant ces bases sont construits à l'aide de termes du langage. Pour des facilités d'expressions et de manipulations, il est préférable de parler de base de connaissances du domaine (BCD) et de base de connaissances du problème (BCP). Chacune d'elles peut contenir des informations sous forme de faits et de règles (1).

Dans le système CRIQUET, il y a une seule base de règles et base de faits courante. Mais il est possible de les sauvegarder dans des fichiers de noms donnés par l'utilisateur (commandes sauvbr, sauvbf). Le nombre de ces fichiers est illimité. Le chargement d'un fichier de règles (d'un fichier de faits) dans la base de règles (dans la base de faits) est obtenu avec la commande "chargbr" ("chargbf") en donnant le nom du fichier voulu. Une base de faits ou de règles peut être construite par la réunion de plusieurs fichiers (ruf, rur). Ainsi une base de connaissances du domaine peut être constituée d'un fichier de règles et d'un fichier de faits chargés lors de chaque référence au domaine. La base de connaissances du problème est constituée de règles et de faits que l'on rajoute aux bases courantes contenant la BCD.

Les commandes "brs" et "bfs" permettent de lister le contenu d'un

fichier de faits ou de règles. La modification du contenu d'un tel fichier ne peut être faite qu'en le chargeant dans la base courante concernée. Ces possibilités facilitent la mise au point d'une base de règles en permettant l'utilisation de plusieurs bases intermédiaires.

1-2/ Types de connaissances exprimables

Trois types de connaissances sont utiles et peuvent être exprimées avec le langage proposé :

- les connaissances sur le domaine, qui correspondent à l'expertise à proprement parler spécifiée à l'aide de règles de production.
- la description du problème à traiter qui constitue un ensemble de connaissances volatiles sous forme de faits.
- des dépendances sémantiques, conditionnelles ou pas, entre les relations utilisées pour décrire le domaine et le problème.

1-3/ Le langage d'expression

Le langage utilisé est de type calcul des prédicats du premier ordre. Des variables peuvent donc être utilisées dans l'expression des connaissances. Le langage est défini par un vocabulaire et une syntaxe :

a/ Le vocabulaire

Le vocabulaire comprend :

- des noms de prédicats d'arité 3 au plus.
- des noms de procédures externes, définies dans l'environnement de travail Lisp.
- des noms de variables commençant par ? : ?X, ?Y
- des noms de constantes.

b/ La syntaxe

Les termes du langage sont des quadruplets de la forme :
(objet prédicat attribut valeur)

L'utilisateur dispose d'un module d'interface pour l'aider à rentrer des connaissances selon un tel format. Ainsi la phrase "les isophotes sont de formes tourmentées" ou "la forme des isophotes est tourmentée" est traduite par le module en :
(isophote être forme tourmenté)

Le langage décrit ici permet l'utilisation de la négation. La phrase "les isophotes ne sont pas de formes tourmentées" est traduite en :

non(isophote être forme tourmenté)

Mais le quantificateur existentiel () et le connecteur ou () ne peuvent pas être utilisés. $A \text{ ou } B \rightarrow C$ s'exprime par $A \rightarrow C$ et $B \rightarrow C$ mais $A \rightarrow B \text{ ou } C$ ne peut s'exprimer que par $A \text{ et non } B \rightarrow C$ et $A \text{ et non } C \rightarrow B$ ce qui complique les spécifications.

I-4/ Les faits et la base de faits

a/ Les faits

Ils peuvent être une connaissance du domaine ou une donnée particulière du problème et n'appartenir alors que temporairement à la base de faits. Un fait s'écrit avec un terme du langage, c'est à dire un quadruplet. Un coefficient de vraisemblance compris entre 0 et 1 est associé à chaque terme constituant un fait pour permettre un raisonnement approximatif. Un exemple de fait est :

((isophote être forme tourmenté) 0.8)

Le coefficient exprime l'importance du fait ou la confiance qu'a l'utilisateur en la réalisation de ce fait.

b/ La base de faits

Elle est constituée de faits et fournit les données particulières du problème. Différents utilitaires permettent de la gérer (voir annexes). Elle peut être construite par réunion de fichiers de faits. Des règles d'incohérence servent à assurer la cohérence sémantique des éléments la constituant.

?SYSTEME : bf

BASE DE FAITS

animal avoir poil COEFF : 0.8

animal manger viande COEFF : 0.6

animal avoir couleur vive COEFF : 0.6

animal avoir tache noir COEFF : 0.5

sujet être taxinomie animaux COEFF : 1.0

I-5/ Base de règles

Des règles de production sont utilisées. Le moteur n'étant pas limité aux clauses de Horn, les règles peuvent avoir une ou plusieurs conclusions liées par un opérateur ET. Les règles sont de la forme :

```
(type (identificateur(n))
      (si (condition-1)
          et (condition-2)
          .
          et (condition-p))
      (alors (action-1)
            et (action-2)
            .
            et (action-q))
      (coefficient (c))
```

où les (condition) sont des termes comme ils ont été définis précédemment ou des conditions sur les variables utilisées. Ces expressions portant sur les variables permettent de spécifier des restrictions à respecter lors de la phase de filtrage.

Exemple :

```
si (isophote être forme tourmenté)
et ( T1 < ?X)
et ( T2 < ?Y)
et ( condition ( ?X < ?Y))
```

On a ici 2 variables (?X et ?Y) et une condition sur ces 2 variables.

Les (action) peuvent être des ajouts ou des suppressions de faits, des suppressions de buts partiels (voir I-6-2 c) ou encore l'appel d'une fonction définie dans l'environnement de travail. Dans chacun de ces cas le format d'écriture est le suivant :

```
(ajoutfait ('terme'))
(suppresfait ('terme'))
(suppresbut ('terme'))
(fonction ('nom'))
```

Les règles, comme les faits, sont pondérées d'un coefficient de plausibilité compris entre 0 et 1.

Le système est "ouvert" du fait de l'utilisation possible de règles entraînant le déclenchement d'un événement extérieur au système grâce à une action du type appel à une fonction. Avec les actions possibles pour les règles, divers effets de bords sont possibles.

L'utilisateur peut typer les règles. La distinction est spécifiée dans l'attribut "type" de chaque règle. Il peut ainsi partitionner la base de règles et organiser les connaissances comme il le désire. Cette organisation peut ensuite être utilisée à l'aide de méta-règles (voir I-7). Il existe par défaut deux types de règles : d'inférence et d'incohérence. Dans une règle d'incohérence les conclusions se résument au mot "erreur". Ces règles permettent d'assurer la compatibilité sémantique, conditionnelle ou non, des faits (voir I-9).

Un exemple :

```
    si (équation être type non-linéaire)
      et (identification être méthode mco)
    alors( erreur)
```

Des utilitaires facilitent la gestion de la base de règles (voir annexes). La commande "ajr" permet l'ajout, en interactif, d'une nouvelle règle. Lors de l'utilisation de cette commande, le système maintient à jour l'ensemble des termes conclusion qui peuvent être utilisés comme but à diagnostiquer lors d'un cheminement arrière.

Exemples :

```
REGLE taxan identificateur (7.)
SI ?animal etre mammifere
ET ?animal avoir sabot
ALORS ajoutfait (?animal etre ongule)
COEFF 0.8
```

```
REGLE taxan identificateur (12.)
SI ?animal etre ongule
ET ?animal avoir rayure noir
ALORS ajoutfait (?animal etre zebre)
COEFF 0.6
```

```
REGLE coherence identificateur (18.)
SI ?animal voler
ET ?animal etre autruche
ALORS erreur
COEFF 1.
```

I-6/ Mécanisme d'exploitation

I-6-1/ Le filtrage

Le filtrage consiste à mettre en correspondance 2 formes : le filtre et le fait. Ce processus permet de déterminer l'ensemble des conflits, constitué de l'ensemble des règles activables dans la situation courante.

Afin de faciliter l'écriture des règles, certains opérateurs de filtrage ont été définis :

- l'opérateur ? filtre n'importe quelle séquence de termes non vide

- les prédicats $<$, $>$, $<=$, $>=$, $=$ appliqués à des nombres filtrent tout fait numériquement compatible au sens des inéquations.

Le motif $(T < 3)$ filtre par exemple les faits $(T = 0)$, $(T < 1)$, $(T <= -1)$.

L'algorithme utilisé dans la phase de pattern-matching, ou filtrage, est optimisé. Il ne procède pas par recherche exhaustive mais par essais successifs et retour arrière en cas d'échec.

I-6-2/ Modes de raisonnement

Trois types principaux de mécanismes d'inférence sont utilisés : le chaînage avant et le chaînage arrière comme dans la plupart des systèmes (13) et le chaînage mixte.

a/ Le chaînage avant :

On cherche à déclencher les règles dont les prémisses sont vérifiées dans la base de faits. Une exécution a toujours lieu suivant une séquence de cycles élémentaires, semblables les uns aux autres, dont la forme est :

Le processus s'arrête quand on a atteint un but cherché ou si plus aucune action ne peut être déclenchée.
Le chaînage avant est déclenché par la commande "dd". Un but peut être spécifié au système avec des variables s'il y a lieu.

Le système tente alors de démontrer qu'un fait Z est vérifié en le considérant comme un but à atteindre. Une règle est applicable lorsque sa partie droite représente un sous but visé. L'appliquer consiste à demander de valider ses prémisses, au besoin en engendrant de nouveaux sous buts.

Lorsque le système n'a pas suffisamment d'informations pour déclencher une règle et qu'aucun but ne peut être atteint, il interroge l'utilisateur et lui demande de valider un fait si possible.

Le cheminement arriére est naturel si l'on a un ou plusieurs buts à vérifier. Il facilite la tâche de description du problème de part les interrogations apportées sur des faits susceptibles d'affiner la description du problème traité.

Enfin, si l'utilisateur ne fournit pas de but particulier, le système dispose d'un ensemble de buts possibles et tente de les diagnostiquer séquentiellement jusqu'à la confirmation de l'un d'eux.

Le chaînage arrière est activé avec la commande "dq".

BASE DE FAITS

```

animal  manger viande COEFF : 0.6
animal  avoir couleur vive COEFF : 0.6
animal  avoir tache noir COEFF : 0.5
sujet   etre taxinomie animaux COEFF : 1.0

```

?SYSTEME : dg
VOULEZ-VOUS PASSER PAR L INTERFACE LANGAGE NATUREL ? oui-non
? n
SI VOUS VOULEZ DIAGNOSTIQUER UN FAIT PARTICULIER
DONNEZ SON TEXTE
SINON RENVOYEZ /;
? ;
LE FAIT SUIVANT EST IL VRAI? oui-non
animal avoir poil
? ?
J ESSAYE LA REGLE :
REGLE taxon identificateur (1.)
SI animal avoir poil
ALORS ajoutfait (animal etre mammifere)
COEFF 0.5

VRAI ? oui-non
? oui
DONNEZ UN COEFF
? 0.8
LE FAIT SUIVANT EST IL VRAI? oui-non
animal donner lait
? non
IL EST VRAI QUE : animal etre guepard

c/ Le chaînage mixte :

En cheminement avant l'utilisateur ne connaît pas le but à atteindre. Mais il peut dans certains cas donner au système une liste de buts intermédiaires. Ce sont des faits qui devront, selon l'avis de l'utilisateur, apparaître en cours de raisonnement.

Ces indications permettent de focaliser le raisonnement vers la résolution de buts donnés et ainsi de pratiquer un cheminement mixte, proche de ce qui est fait dans TANGO (1). A chaque cycle on obtient en phase de détection un premier ensemble de règles candidates.

Le système utilise alors les buts partiels, en tenant compte ou pas de l'ordre dans lequel ils ont été donnés. L'utilisateur dispose d'une commande pour préciser ce point.

Dans le premier cas, le système valide si possible parmi les règles candidates celles qui contiennent en conclusion le but courant. Ce dernier est le premier dans la liste des buts partiels. Il n'est enlevé de la liste que lorsque une règle contenant l'action "suppresbut" est activée.

Dans le deuxième cas, le système travaille de la même façon mais en considérant tous les buts partiels restants. Un but partiel est éliminé si il apparaît dans la base de faits.

Si aucune règle ne peut être ainsi validée le cycle se déroule en chaînage avant classique.

Cette technique permet de restreindre à certains moments l'espace de recherche et de réaliser des plans partiels de résolution à chaque cycle du moteur (1).

Le chaînage mixte est utilisé à chaque appel du chaînage avant avec une base de buts non vide. Des utilitaires permettent de gérer la base de buts (voir annexes). Les commandes "avecbutord" et "sansbutord" permettent d'indiquer au système s'il faut respecter ou pas l'ordre dans lequel les buts partiels ont été donnés. La base de buts est gérée en mode LIFO. Le dernier but rentré lors de la création de la base sera le premier considéré par le système.

I-6-3/ Choix d'une règle

Il s'agit de choisir la règle à activer parmi l'ensemble des règles candidates et ainsi de résoudre les conflits existant si il y a lieu. La stratégie de choix est vitale non seulement pour les performances mais également pour la capacité du système à comprendre ce qu'il fait. Il faut déterminer les critères qui rendent une règle plus prioritaire que les autres.

Il existe trois grandes familles de structures de contrôle qui sont : choix par évaluation, recherche exhaustive et contrôle par méta-règles.

Dans le premier cas, une méthode est de prendre la première règle activable. C'est le critère le plus simple mais dépendant de l'ordre d'écriture des règles. On peut aussi choisir la règle la plus récemment utilisée ou encore la règle la plus contraignante, c'est à dire celle qui possède le plus grand nombre de termes condition. Dans ce système, deux méthodes sont proposées :

- choix de la règle de plus forte vraisemblance
- choix de la règle utilisant les faits les plus récents

Dans le premier cas, on met l'accent sur les déductions. Dans le second, on favorise les objets et les transitions sont fonction des états et non uniquement des actions. De nouvelles fonctions de choix par évaluation peuvent être facilement introduites. La commande "chx" permet de passer d'une fonction à une autre.

Un contrôle par méta-règle peut aussi être utilisé comme dans MYCIN (9 12). Il suffit pour cela de créer des méta-règles spécifiques qui permettent une sélection des règles objets selon leur type ou à l'aide de conditions portant sur les prémisses ou actions de ces règles. La phase de choix débute alors par une recherche d'une méta-règle activable. Celle-ci, si elle existe, fournit un critère de sélection parmi les règles objets candidates.

I-7/ La méta-connaissance

La méta-connaissance est une donnée qui permet au système d'améliorer ses performances et son efficacité.

Pour améliorer de façon sensible les performances, en particulier dans le processus de filtrage, deux solutions particulières sont possibles : soit modifier le processus de filtrage (1), soit réduire l'ensemble des règles à manipuler. Des méta-règles sont utilisées dans ce cas. Ces éléments introduits par R. Davis (3) sont des règles qui agissent sur des règles objets. Les méta-règles font appel aux règles objet par leur contenu et non par leur nom. Ces outils ont déjà été utilisés dans beaucoup de systèmes tel MYCIN (9).

Dans le système proposé des outils sont disponibles pour construire et utiliser de telles méta-règles. Celles-ci permettent de sélectionner des règles par une recherche sur leur type ou sur un ou plusieurs mots ou termes dans les prémisses et conclusions (commandes "rht" "rhp" "rha" "rhfp" "rhfa").

Exemple d'une méta-règle :

```
?SYSTEME : bmr
REGLE meta identificateur (1.)
SI sujet etre taxinomie animaux
ALORS rht (taxan)
COEFF 1.
```

```
REGLE meta identificateur (2.)
SI ?animal avoir poil
ET ?animal manger viande
ALORS rha (?animal etre carnivore)
ET rhp (?animal etre carnivore)
ET rhp (?animal avoir poil)
COEFF 0.8
```

L'utilisateur peut indiquer avec les commandes "avecmeta" et "sansmeta" s'il veut travailler avec ou sans les méta-règles pour restreindre l'espace de recherches et ce, quelque soit le type de cheminement utilisé.

Les méta-règles peuvent aussi être intégrées aux structures de contrôle (I-6-3). Des heuristiques de guidage de la recherche agissent à travers ces méta-règles. Déjà, l'utilisation de buts partiels, comme cela a été décrit précédemment, permet de guider les recherches du système et ainsi d'améliorer son efficacité. La fonction "chx" permet aussi d'obliger le système à utiliser un contrôle par méta-règles, en indiquant la fonction de choix par évaluation à utiliser dans le cas où aucune méta-règle ne peut

être activée.

Ainsi, la structure de contrôle et les heuristiques sont mises en évidence au lieu d'être noyées dans le corps du programme (3).

1-8/ Le raisonnement approximatif

Les règles de production ne reflètent pas des implications logiques mais plutôt les convictions de l'expert. Les règles comme les faits constituant l'univers sont valués avec un coefficient de vraisemblance compris entre 0 et 1. Lors d'une déduction, la méthode de calcul d'une valuation pour un fait déduit est proche de celle utilisée dans MYCIN (11).

Soit la règle R1 : SI A1 et A2 et ... An ALORS R le coefficient du fait déduit B est donné par :

$$cv(B) = cv(R1) * \text{Min } cv(Ai)$$

Nous n'avons pas voulu utiliser de concepts probabilistes du fait de la non vérification de certaines hypothèses mathématiques. Ainsi la non indépendance de termes condition de règles rend l'utilisation de tels concepts peu adaptée. Il a déjà été reconnu (7) que l'utilisation privilégiée de max et min dans les fonctions de composition empêche tout effet de renforcement ou d'affaiblissement, alors que ces effets sont fondamentaux dans la représentation de l'évolution de notre jugement. Pour répondre en partie à cette critique un mécanisme de confirmation d'un fait a été mis en place. Si la règle activée confirme avec le coefficient cv2 un fait déjà présent dans la base de faits avec coefficient cv1, le coefficient renforcé est alors égal à : $cv = cv1 + cv2 - cv1*cv2$. Un tel mécanisme est déjà utilisé dans des systèmes tels MYCIN et SNARK (11) (8). L'élaboration d'un mode performant de combinaison des coefficients n'est pas notre but et nous nous sommes contentés des mécanismes décrits ci-dessus. Dans le domaine de l'étude du raisonnement approximatif en général, de nombreux problèmes restent à résoudre (7) (10) (11).

Dans le moteur proposé, les règles d'incohérence sont valuées et manipulées comme les règles d'inférence. Le système peut ainsi tenir un raisonnement sans rejeter toutes les incohérences et ambiguïtés existantes. En effet, à l'aide de la valuation des règles le concepteur peut faire en sorte que certaines inférences soient possibles même en présence d'incohérences. Il suffit pour cela que les coefficients de vraisemblance de certaines règles d'inférence soient plus forts que ceux des règles d'incohérence candidates. Il est nécessaire alors de travailler avec la première fonction de choix citée précédemment. Ces possibilités sont intéressantes pour se rapprocher du raisonnement humain mais

rendent très importante l'évaluation correcte des pondérations des règles.

1-9/ Gestion de compatibilité

Un raisonnement basé sur des coefficients de plausibilité peut empêcher de discerner, parmi les conclusions portant sur une même entité, les hypothèses concurrentes des composantes d'un même résultat. La gestion de compatibilité proposée permet de valider et d'affiner le raisonnement tenu par le système. Le système maintient, dans la limite de ses connaissances, la compatibilité logique et sémantique des faits.

La vérification de la compatibilité logique ne consiste qu'en la reconnaissance de la négation d'un fait lors d'ajouts de nouveaux faits. On ne peut pas ajouter la négation d'un fait déjà présent. Il est possible ainsi de démontrer la négation d'un fait.

La compatibilité sémantique est assurée à l'aide de contraintes contextuelles spécifiées par l'utilisateur avec des règles d'incohérence proches des méthodes utilisées dans MIRLITHO (4). La commande "coh" permet de spécifier une liste de faits toujours incompatibles avec un fait particulier. On donne pour un fait F1 une liste de faits (F2...fn). Le système génère alors (n-1) règles d'incohérence telles que : si (F1 et F2) alors (erreur). Des incohérences conditionnelles peuvent aussi être données. Si les faits X et Y sont incompatibles en présence des faits A et B il suffit de créer une règle d'incohérence avec pour prémisse : (X et Y et A et B).

Cette gestion de cohérence est faite, si l'utilisateur le désire, lors d'ajouts de faits et lors de chaque phase de déduction dans le raisonnement.

Exemple : gestion à la création de la base de faits

?SYSTEME : avecgsem

?SYSTEME : prcoh

REGLES DE COHERENCE :

REGLE coherence identificateur (17.)

SI ?animal avoir poil

ET ?animal avoir plume

ALORS erreur

COEFF 1.

REGLE coherence identificateur (18.)

SI ?animal voler

ET ?animal etre autruche

ALORS erreur

COEFF 1.

CRIQUET

?SYSTEME : cbf
VOULEZ-VOUS PASSER PAR L'INTERFACE LANGAGE NATUREL ? oui-non
? 0
POUR SIGNALER LA FIN DES DONNEES TAPPEZ /;
DONNEZ UN FAIT
? 1`animal vole;

PHRASE CORRECTE VIS A VIS DU CONTEXTE
animal voler
ETES-VOUS D'ACCORD AVEC CETTE TRADUCTION ?
? 0
DONNEZ UN COEFF
? 0.8
DONNEZ UN FAIT
? 1`animal est une autruche;

PHRASE CORRECTE VIS A VIS DU CONTEXTE
animal etre autruche
ETES-VOUS D'ACCORD AVEC CETTE TRADUCTION ?
? 0
attention : ce fait est incoherent avec le reste de la BF
a cause de la regle :
REGLE coherence identificateur (18.)
SI animal voler
ET animal etre autruche
ALORS erreur
COEFF 1.

DONNEZ UN FAIT
? ;
VOILA COMMENT J'AI COMPRIS LA BASE DE FAITS
BASE DE FAITS
animal voler COEFF : 0.8
ETES VOUS ACCORD ? oui-non
? 0

Les règles d'incohérence sont évaluées et manipulées par le système comme les règles d'inférence (voir I-8). On peut éprouver la validité de la base de règles en détectant des résultats contradictoires fournis sur un jeu de données valides (4).

Une règle d'incohérence activée devra par la suite non pas arrêter le raisonnement, mais signaler l'incohérence et déclencher un retour-arrière selon une technique choisie.

Nous avons essayé de faciliter au mieux les communications homme-machine. Il est primordial que, quelque soit le domaine d'application, le système soit d'accès aisé aux utilisateurs, que ces derniers aient confiance et puissent vérifier les mécanismes de raisonnement utilisés. Ceci est fondamental à la fois dans la phase d'utilisation et de développement du système.

Comme la plupart des systèmes experts, le mécanisme peut expliquer son raisonnement que ce soit lors d'un cheminement avant ou arrière avec la commande "expl". De plus, toutes les commandes sont sans paramètres. Les éléments nécessaires au bon fonctionnement sont demandés par le système au moment voulu. Les ajouts et modifications de règles et de faits en sont facilités.

Un module simple d'interface rend plus aisée la donnée des faits.

Lors d'un cheminement arrière, l'utilisateur peut demander la justification d'une question posée par le système ("?").

En chaînage avant, un travail en mode "trace" est possible et utile en particulier pour la mise au point des règles. Sous ce mode de travail, lors de chaque phase de choix, le système expose à l'utilisateur quelles sont les règles activables et quel est son choix. L'utilisateur peut l'approuver ou le rejeter. Dans ce dernier cas le système répète une opération de choix sur les règles activables restantes. Si il n'en reste plus le système utilise la règle choisie en dernier lieu.

L'utilisateur peut aussi demander l'explication du raisonnement tenu jusqu'alors ("expl") ou reinitialiser le dernier cycle ("rappel") en changeant si il le désire la fonction de choix à utiliser ("chx"), ou en ajoutant des éléments dans la base de faits ("ajf"). Ces fonctionnalités sont utiles pour le développement des bases de connaissances et la mise au point d'une pondération correcte des règles.

Dans tout travail d'élaboration d'un système la mémorisation d'un historique de travail est nécessaire. La commande "sauve" appelée en début de session permet, une fois sorti du système avec la commande "fin", de garder une trace de toutes les opérations faites durant la dernière session de travail. La commande "fin", au lieu de "end", permet de sortir du système en sauvegardant toutes les modifications faites dans les bases de règles et de faits.

II/ LE MODULE D'INTERFACE

II-1/ Introduction

Le module d'interface facilite la spécification des informations, en évitant à l'utilisateur d'avoir à formaliser ses connaissances selon le format interne strict utilisé par le système. L'interface tente de traduire une phrase simple en une représentation interne. Le vocabulaire employé dans les faits reste compatible avec celui des règles. Sans le module d'interface, des erreurs de spécifications sont possibles en particulier à cause des synonymes naturellement utilisés par l'homme mais méconnus du système. L'interface permet aussi d'enrichir le vocabulaire du système et ainsi sa connaissance.

II-2/ L'interface construite

Le traitement du langage naturel est un problème abordé en Intelligence Artificielle. Le but ici n'est pas aussi ambitieux. La grammaire reconnue est simple. Les phrases traitées doivent être suffisamment courtes pour pouvoir être représentées à l'aide du format :

< objet prédicats attribut valeur >

Pour traiter chaque phrase, le système procède à une analyse lexicale, une analyse syntaxique et à des vérifications sémantiques.

Des dictionnaires de noms, de verbes et d'adjectifs sont utilisés. Des commandes permettent de compléter ou de modifier ces dictionnaires.

Pour chaque phrase, le système s'il ne détecte pas d'erreur propose sa traduction à l'utilisateur. Ce dernier peut s'il n'est pas d'accord reformuler son information.

II-3/ Principes de fonctionnement

La traduction se fait en deux passages. Le premier permet la suppression des mots vides tels que les articles, les adverbes et les prépositions. Le système dispose là aussi d'un dictionnaire de mots vides. Au deuxième passage, une analyse lexicale et syntaxique permettent de construire la représentation interne selon le format :

< nom > < verbe > < nom > < adjectif >

II-3-1/ Analyse lexicale

L'analyse lexicale rend le type de chaque terme et le mot significatif associé. Il y a trois types de données reconnus : les noms, les verbes et les adjectifs. A chacun est associé un dictionnaire. Les dictionnaires sont des listes de la forme : ((mot-significatif type synonyme1 synonyme2 ...)(...) ...)
Le type des noms est : attribut (\$a), objet (\$o) ou neutre (\$ao), celui des verbes est : transitif ou intransitif. Les adjectifs ne sont pas typés. Les mots significatifs sont ceux utilisés dans la représentation interne. Si un mot n'apparaît dans aucun des dictionnaires, il est traité comme un mot inconnu.

II-3-2/ Analyse syntaxique

Deux classes de phrases sont reconnues en entrée, qui sont les formes actives et les formes passives. Le résultat de la traduction est toujours une forme active. Dans tous les cas, la grammaire reconnue est simple.

Forme active

- <nom><forme active transitive>(<nom><adj>)/(<adj><nom>)
exemple : "le chat mange la souris grise" est traduit en
(chat manger souris grise)

- <nom1><nom2><etre><adjectif>
Le complément du nom est traité et la traduction est :
(nom2 avoir nom1 adjectif)
exemple : "le chat du voisin est gris " est traduit en
(voisin avoir chat gris)

- <nom><forme active intransitive>
exemple : "mon voisin tombe " est traduit en
(voisin tomber)

Forme passive

- (<adjectif><nom>)/(<nom><adjectif>) <forme passive><nom>
exemple : "la petite souris est mangée par le chat" est traduit en
(chat manger souris petite)

Une forme passive est obligatoirement de la forme :

- <etre><verbe transitif>

La négation est reconnue par la présence simultanée des mots "ne" et "pas" ou par la présence d'un verbe préfixé par "n'" et du mot "pas".

II-3-3/ Vérifications sémantiques

Une analyse syntaxique qui s'achève correctement rend une expression de la forme : <nom1><verbe><nom2><adjectif>. Les éléments <nom1> et <nom2> possèdent un type, précisé dans le dictionnaire des noms, qui a été fixé par l'utilisateur selon le contexte de travail. Les trois types possibles sont : objet, attribut et neutre. Il y a erreur sémantique si la traduction rendue ne vérifie pas le format

<objet><prédicat><attribut><valeur>

c'est à dire si le premier nom n'est pas un objet ou neutre et le second nom n'est pas un attribut ou neutre.

II-4/ Enrichissement du vocabulaire et traitement des

mots inconnus

Des commandes ("prd ajmd spmd mdm") permettent de modifier et de compléter les dictionnaires indépendamment de tout autre travail. Lors d'ajouts de règles ou de faits, si les mots utilisés ne sont pas tous connus, CRIQUET dialogue avec l'utilisateur. Le système l'interroge pour apprendre le type du mot concerné, savoir si c'est un synonyme d'un mot significatif déjà connu ou pas. Les dictionnaires sont complétés comme il le faut selon les cas. Ce n'est que lors d'ajouts de règles dans la base de connaissances que de nouveaux mots significatifs peuvent être introduit. De cette façon le vocabulaire des faits et des règles reste compatible.

CONCLUSION

Le moteur CRIQUET est encore en développement, en particulier en vue d'établir une meilleure gestion de la cohérence de la base de faits et une gestion des échecs, avec retour arrière s'il y a lieu. Le module d'interface peut aussi être amélioré en accroissant le nombre d'éléments qui composent les termes du langage et en élargissant la grammaire reconnue. Enfin, pour améliorer les performances du système, la base de règles pourra être compilée (1).

CRIQUET est élaboré au sein d'un projet plus important, EDORA (5), développé au centre INRIA de Sophia Antipolis. Le sujet de ce projet concerne la modélisation des connaissances à l'aide d'objets dynamiques proches des frames, pour lesquels la manipulation de règles de production est aussi nécessaire.

REMERCIEMENTS

Je tiens à remercier C.Ganger, INRIA Sophia Antipolis, ainsi que O.Corby, M.Montalban, P.Redino, étudiants du DESS "Informatique et Sciences de l'Ingénieur", pour leur aide apportée dans le développement de ce moteur.

ANNEXES

Les commandes utilisables sont listées avec pour chacune une définition sommaire de leur fonctionnalité.

?SYSTEME : menu

MENU

liste des commandes disponibles

prd : pour imprimer le contenu d un dictionnaire du module d'interface

ajmd : pour ajouter un mot dans un dictionnaire de l interface

spmd : pour supprimer un mot dans un dictionnaire

mdm : pour modifier le type d un nom dans le dictionnaire des noms

ajf : ajouter des faits en conversationnel dans la base de faits

spf : supprimer un fait dans la base de faits

mdf : modifier le coeff de plausibilite d un fait

cbf : construire la base de faits

spbf : detruire la base de faits

bf : imprimer la base de faits

bh : imprimer l ensemble hypotheses

ajb : pour ajouter un but dans buts

spb : pour supprimer un but dans buts

cbb : pour creer l ensemble buts

spbb : pour detruire tout buts

bb : pour imprimer buts

avecbutord : pour tenir compte de l ordre des buts partiels

sansbutord : pour ne pas tenir compte de leur ordre

pr : impression d une regle

prl : impression d une liste de regles

prcoh : impression des regles de coherence connues

prc : impression de regles selon une recherche sur le coefficient de vraisemblance

br : impression de la base de regles

mdr : pour supprimer ou ajouter des premisses ou modifier le coefficient de plausibilite

spr : suppression d une regle

ajr : ajout d une regle

spbr : supprimer toute la base de regles

coh : specifier une liste de faits incompatibles avec un fait

avecglg : pour assurer la coherence logique lors des deductions

avecglsem : pour assurer la coherence semantique lors des deductions

sansglg : sans la gestion de coherence logique

sansgsem : sans la gestion de coherence semantique

rhp : recherches de regles sur un ou plusieurs mots dans les premisses

rha : recherches de regles sur un ou plusieurs mots dans les actions

rhfa : recherche d'une forme dans les parties actions

rhfp : recherche d'une forme dans les premisses

rht : recherche des regles selon leur type

rhc : recherche des regles de coherence

rhi : recherche des regles d'inference

rst : restriction de la base de regles utilisees

brs : pour lister le contenu d'un fichier de regles

bfs : pour lister un fichier de faits

pmr : impression d'une meta-regle

bmr : impression de la base de meta-regles

ajmr : ajout d'une meta-regle

spm : suppression d'une meta-regle

avecmeta : pour travailler en utilisant les meta-regles

sansmeta : pour ne plus utiliser les meta-regles

pnbf : lister les noms utilises pour sauvegarder des BF

pnbr : lister les noms utilises pour sauvegarder des BR

dd : activer le raisonnement en chainage avant

dg : activer le raisonnement en chainage arriere

dem : activer le raisonnement en chainage mixte

chx : pour choisir la strategie de choix

expl : pour avoir les explications sur le raisonnement

resbr : restaurer l'ancienne base de regles

resbf : recupere la derniere base de faits

sauvbf : sauvegarder la base de faits courante

sauvbr : sauvegarder la base de regles courante

chargbf : charger une nouvelle base de faits

chargbr : charger une nouvelle base de regles

ruf : pour faire la reunion d'un fichier de faits avec la base de faits courante

rur : pour faire la reunion d'un fichier de regles avec la base de regles courante

avectrace : travailler avec traces

sanstrace : travailler sans traces - mode adopte par default

info : pour avoir des informations sur le systeme

help : pour avoir des informations sur une commande en particulier

sauve : pour garder une trace de toute la session

fin : pour sortir du systeme en sauvegardant l'environnement

end : pour sortir du systeme sans rien sauvegarder

essai : pour avoir une petite base de regles pour des essais

Exemple : ajout d'une règle

?SYSTEME : ajr
VOULEZ-VOUS PASSER PAR L INTERFACE LANGAGE NATUREL ? oui-non
? o
DONNEZ LE TYPE DE REGLE ?
PAR DEFAUT : i POUR INFERENCE
ip POUR INFERENCE AVEC DES CONDITIONS SUR LES VARIABLES
c POUR COHERENCE
? taxan
DONNER UNE CONDITION- /; SEUL EN FIN
? ?animal est un oiseau;

PHRASE CORRECTE VIS A VIS DU CONTEXTE
?animal etre oiseau
ETES-VOUS D'ACCORD AVEC CETTE TRADUCTION ?
? o
DONNER UNE CONDITION- /; SEUL EN FIN
? ?animal est maritime;
LE MOT SUIVANT EST INCONNU :

maritime

INDIQUER SI C EST UN NOM <TAPER:nom>
OU UN VERBE <TAPER:verbe>
OU UN ADJECTIF <TAPER:adj>
OU UNE FORME CONJUGUEE DU VERBE ETRE:<TAPER:etre>
OU UN MOT VIDE < EX: ARTICLE ADVERBE PREPOSITION ETC>
DANS CE CAS :<TAPER:ptmot>
SI C EST AUTRE CHOSE:<TAPER:q>
? adj
EST CE QUE CE MOT EST UN SYNONYME D UN DE CEUX QUI SUIVENT ?
bleu chaud jaune gris carnivore pointu aigu ongule vive
noir long
? n

PHRASE CORRECTE VIS A VIS DU CONTEXTE
?animal etre maritime
ETES-VOUS D'ACCORD AVEC CETTE TRADUCTION ?
? o
DONNER UNE CONDITION- /; SEUL EN FIN
? ;
DONNER UNE ACTION- /; SEUL EN FIN
? ajoutfait : ?animal est une mouette;
LE MOT SUIVANT EST INCONNU :

mouette

INDIQUER SI C EST UN NOM <TAPER:nom>

OU UN VERBE <TAPER:verbe>

OU UN ADJECTIF <TAPER:adj>

OU UNE FORME CONJUGUEE DU VERBE ETRE:<TAPER:etre>

OU UN MOT VIDE < EX: ARTICLE ADVERBE PREPOSITION ETC>

DANS CE CAS:<TAPER:ptmot>

SI C EST AUTRE CHOSE:<TAPER:q>

? nom

EST CE QUE CE MOT EST UN SYNONYME D UN DE CEUX QUI SUIVENT ?

animal lapin souris guepard pingouin autruche zebre girafe tigre

plume poil oeuf mammifere lait oiseau dent griffe regard viande

tache rayure jambe albatros sabot couleur

? n

EST CE UN ATTRIBUT <TAPER: \$a

UN OBJET <TAPER: \$o> OU

LES DEUX <TAPER: \$ao>

? \$ao

PHRASE CORRECTE VIS A VIS DU CONTEXTE

?animal etre mouette

EYES-VOUS D'ACCORD AVEC CETTE TRADUCTION ?

? o

DONNER UNE ACTION- /; SEUL EN FIN

? ;

DONNER LE COEFFICIENT DE VRAISEMBLANCE

? 0.8

VOILA COMMENT J AI COMPRIS CETTE REGLE

REGLE taxan identificateur (19.)

SI ?animal etre maritime

ET ?animal etre oiseau

ALORS ajoutfait (?animal etre mouette)

COEFF 0.8

ETES VOUS D ACCORD ? oui - non

? o

VOULEZ-VOUS CONTINUER ? oui-non

? n

Exemple : un cheminement avant en mode trace

?SYSTEME : dd
VOULEZ-VOUS PASSER PAR L INTERFACE LANGAGE NATUREL ? oui-non
? n
SI VOUS RECHERCHEZ UN BUT PARTICULIER
DONNEZ SON TEXTE SINON RENVOYEZ /;
? i
LES REGLES ACTIVABLES SONT :
REGLE taxan identificateur (1.)
SI animal avoir poil
ALORS ajoutfait (animal etre mammifere)
COEFF 0.5

REGLE taxan identificateur (5.)
SI animal manger viande
ALORS ajoutfait (animal etre carnivore)
COEFF 0.9

AVEC LA BASE DE FAITS :
BASE DE FAITS

=====

animal	avoir poil	COEFF : 0.8
animal	manger viande	COEFF : 0.6
animal	avoir couleur vive	COEFF : 0.6
animal	avoir tache noir	COEFF : 0.5
sujet	etre taxinomie animaux	COEFF : 1.0

J AI CHOISIT LA REGLE :
REGLE taxan identificateur (5.)
SI animal manger viande
ALORS ajoutfait (animal etre carnivore)
COEFF 0.9

ETES VOUS D ACCORD ? oui-non
? chx

INDIQUER LA STRATEGIE A UTILISER 1. ou 2.
1. : CHOISIR LA REGLE DE PLUS FORT COEFFICIENT
2. : CHOISIR LA REGLE QUI UTILISE LES FAITS RECENTS
3. : CONTROLE PAR META-REGLES
VOUS UTILISEZ EN CE MOMENT LA STRATEGIE NUMERO : 1.
? 2

LES REGLES ACTIVABLES SONT :
REGLE taxan identificateur (1.)
SI animal avoir poil
ALORS ajoutfait (animal etre mammifere)
COEFF 0.5

REGLE taxan identificateur (5.)
SI animal manger viande
ALORS ajoutfait (animal etre carnivore)
COEFF 0.9

AVEC LA BASE DE FAITS :
BASE DE FAITS

=====

animal	avoir poil	COEFF	: 0.8
animal	manger viande	COEFF	: 0.6
animal	avoir couleur vive	COEFF	: 0.6
animal	avoir tache noir	COEFF	: 0.5
sujet	etre taxinomie animaux	COEFF	: 1.0

J AI CHOISIT LA REGLE :
REGLE taxan identificateur (1.)
SI animal avoir poil
ALORS ajoutfait (animal etre mammifere)
COEFF 0.5

ETES VOUS D ACCORD ? oui-non
? o

LES REGLES ACTIVABLES SONT :
REGLE taxan identificateur (5.)
SI animal manger viande
ALORS ajoutfait (animal etre carnivore)
COEFF 0.9

AVEC LA BASE DE FAITS :
BASE DE FAITS

animal	etre mammifere	COEFF	: 0.4
--------	----------------	-------	-------

=====

animal	avoir poil	COEFF	: 0.8
animal	manger viande	COEFF	: 0.6
animal	avoir couleur vive	COEFF	: 0.6
animal	avoir tache noir	COEFF	: 0.5
sujet	etre taxinomie animaux	COEFF	: 1.0

J AI CHOISIT LA REGLE :
REGLE taxan identificateur (5.)
SI animal manger viande
ALORS ajoutfait (animal etre carnivore)
COEFF 0.9

LES REGLES ACTIVABLES SONT :
REGLE taxan identificateur (9.)
SI animal etre mammifere
ET animal etre carnivore
ET animal avoir couleur vive
ET animal avoir tache noir
ALORS ajoutfait (animal etre guepard)
COEFF 0.6

AVEC LA BASE DE FAITS :
BASE DE FAITS
animal etre carnivore COEFF : 0.54
animal etre mammifere COEFF : 0.4
=====

animal	avoir poil	COEFF	: 0.8
animal	manger viande	COEFF	: 0.6
animal	avoir couleur vive	COEFF	: 0.6
animal	avoir tache noir	COEFF	: 0.5
sujet	etre taxinomie animaux	COEFF	: 1.0

J. AI CHOISIT LA REGLE :
REGLE taxan identificateur (9.)
SI animal etre mammifere
ET animal etre carnivore
ET animal avoir couleur vive
ET animal avoir tache noir
ALORS ajoutfait (animal etre guepard)
COEFF 0.6

ON OBTIENT LA CONCLUSION SUIVANTE :
animal etre guepard COEFF : 0.24

Exemple : une explication du système après un cheminement avant

?SYSTEME : expl
LE SYSTEME VA EXPLIQUER SON RAISONNEMENT A L AIDE DES
REGLES UTILISEES
LORS D UN CHEMINEMENT AVANT

=====

AVEC LA REGLE identificateur (1.)
REGLE taxan identificateur (1.)
SI animal avoir poil
ALORS ajoutfait (animal etre mammifere)
COEFF 0.5

ET LA BASE DE FAITS :
BASE DE FAITS

=====

animal avoir poil COEFF : 0.8
animal manger viande COEFF : 0.6
animal avoir couleur vive COEFF : 0.6
animal avoir tache noir COEFF : 0.5
sujet etre taxinomie animaux COEFF : 1.0

NOUS POUVONS DEDUIRE
ajoutfait (animal etre mammifere)

AVEC LA REGLE identificateur (5.)
REGLE taxan identificateur (5.)
SI animal manger viande
ALORS ajoutfait (animal etre carnivore)
COEFF 0.9

ET LA BASE DE FAITS :
BASE DE FAITS

animal etre mammifere COEFF : 0.4
=====

animal avoir poil COEFF : 0.8
animal manger viande COEFF : 0.6
animal avoir couleur vive COEFF : 0.6
animal avoir tache noir COEFF : 0.5
sujet etre taxinomie animaux COEFF : 1.0

NOUS POUVONS DEDUIRE
ajoutfait (animal etre carnivore)

AVEC LA REGLE identificateur (9.)
REGLE taxon identificateur (9.)
SI animal etre mammifere
ET animal etre carnivore
ET animal avoir couleur vive
ET animal avoir tache noir
ALORS ajoutfait (animal etre guepard)
COEFF 0.6

ET LA BASE DE FAITS :

BASE DE FAITS

animal etre carnivore COEFF : 0.54

animal etre mammifere COEFF : 0.4

=====

animal avoir poil COEFF : 0.8

animal manger viande COEFF : 0.6

animal avoir couleur vive COEFF : 0.6

animal avoir tache noir COEFF : 0.5

sujet etre taxinomie animaux COEFF : 1.0

NOUS POUVONS DEDUIRE

ajoutfait (animal etre guepard)

BIBLIOGRAPHIE

- 1 - Cordier MO, Rousset MC
"TANGO : moteur d'inférence pour un système expert
avec variables "
Proc 4ieme congrès AFCET-INRIA
Reconnaisances des formes et intelligence artificielle
25-27 janvier 1984
- 2 - Cordier MO
"Les systèmes experts "
La Recherche n=151 Janvier 1984
- 3 - Davis R
" Meta rules : reasoning about control "
Artificial intelligence journal
Dec 1980 -15- pp 179-222
- 4 - Ganascia JG
"Etude des contradictions entre les données dans les
systèmes de diagnostic"
Proc 4ieme journees congrès AFCET-INRIA
Reconnaisances des formes et intelligence artificielle
25-27 janvier 1984
- 5 - Gouze JL, Vignard P
"EDORA : un système intelligent d'aide à la modélisation
en biologie "
International 84 AMSE Conference "Modelling and Simulation"
Athen (Greece) June 27-29 1984
- 6 - Granger C, Thonnat M, Vignard P
"Etude d'un système expert pour la classification
de galaxies : SYGAL "
Les systèmes experts et leurs applications :
Journées d'étude et exposition
Palais des Congrès - Avignon 2, 3, 4 Mai 1984
- 7 - Kayser D
"Vers une modélisation du raisonnement approximatif "
Proc Colloque "Représentation des connaissances et raisonnement
dans les sciences de l'homme"
Saint-Maximin ed M Borillo pub par INRIA (440-457)
- 8 - Lagrange MS, Renaud M
"Deux expériences de simulation du raisonnement en Archéologie
au moyen d'un système expert : le système SNARK "
Les systèmes experts et leurs applications :
Journées d'étude et exposition
Palais des Congrès - Avignon 2, 3, 4 Mai 1984

- 9 - Lauriere JL
"Représentation et utilisation des connaissances.
Première partie: Les systèmes experts"
TSI Techniques et Sciences Informatiques Vol 1, n=1, 1982
- 10 - Prade H
" Modeles mathematiques de l'imprecis et de l'incertain en vue
d'applications au raisonnement naturel "
Thèse d etat , Toulouse III, Juin 1982
- 11 - Shortliffe E, Buchanan BG
"A model of inexact reasoning in medecine "
Mathematical Biosciences 23, 351-379 (1975)
- 12 - Waterman D, Hayes-Roth (Eds)
"Pattern Directed Inference Systems"
Academic Press N.Y (1978)
- 13 - Winston PH
"LISP"
Addison Wesley Publishing Company 1981

1
2

3
4

5

6

7

8

9

10

11

12

13

14

15